

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-334870

(43)公開日 平成6年(1994)12月2日

(51)Int.Cl.⁵

H 04 N 1/41

G 06 F 15/66

識別記号

庁内整理番号

F I

技術表示箇所

C 9070-5C

3 3 0 C 8420-5L

審査請求 未請求 請求項の数3 FD (全13頁)

(21)出願番号

特願平5-141524

(22)出願日

平成5年(1993)5月19日

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 小松 学

東京都大田区中馬込1丁目3番6号 株式会社リコー内

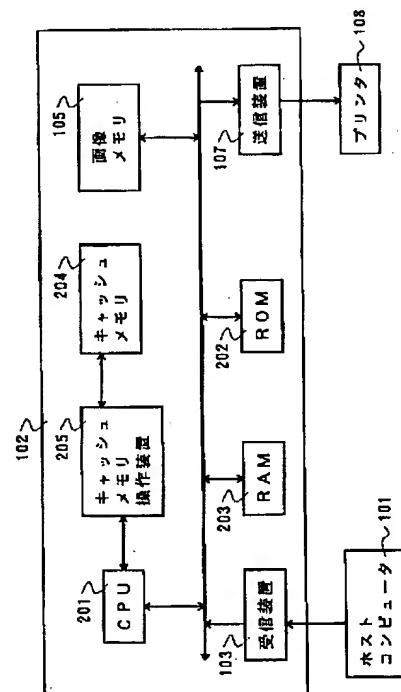
(74)代理人 弁理士 酒井 宏明

(54)【発明の名称】 符号化装置

(57)【要約】

【目的】 従来方式の復号画像と同等の画質を保持し、従来のような無駄な符号データを少なくなくして符号化効率を向上させる。

【構成】 多値(カラー)画像情報をブロックに分割して、該ブロック毎に符号データを構成し、符号化する多値(カラー)画像の符号化装置において、前記ブロック内の画素値の構成情報を抽出し、前記ブロック内構成情報に応じて符号データの構成を変化させ、該符号データの一部であるブロック内の代表値を整数で割った際の余りの数により前記符号データの構成を切り換えるCPU201を具備することを特徴とする。



【特許請求の範囲】

【請求項1】 多値（カラー）画像情報をブロックに分割して、該ブロック毎に符号データを構成し、符号化する多値（カラー）画像の符号化装置において、前記ブロック内の画素値の構成情報を抽出し、前記ブロック内構成情報に応じて符号データの構成を変化させ、該符号データの一部であるブロック内の代表値を整数で割った際の余りの数により前記符号データの構成を切り換える制御手段を具備することを特徴とする符号化装置。

【請求項2】 前記制御手段は、前記ブロック内の全画素値が同一の場合、先のブロック内情報も同一であるか探索し、該情報に応じて複数ブロックを連結して符号化することを特徴とする請求項1記載の符号化装置。

【請求項3】 前記制御手段は、前記ブロック内が2種類の画素値で構成されている場合、画素値情報を符号化するための複数のブロック内量子化値構成パターンを示すコード番号と前記ブロック内の代表値により符号化することを特徴とする請求項1記載の符号化装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、デジタルカラー画像処理に利用され、多値（カラー）画像情報をブロックに分割して、該ブロック毎に符号データを構成し、符号化する多値（カラー）画像の符号化装置に関する。

【0002】

【従来の技術】従来、多値のカラー画像情報をブロックに分割して、該ブロック毎に符号データを構成し、符号化するカラー画像の符号化方法であって、自然画像以外の空間周波数の高いカラー画像であっても復号画像の品質が良い符号化方法として、固定長G B T C (Generalized Block Truncation Coding)型符号化方式が知られている。

【0003】図15は、固定長G B T C型符号方式を示す説明図である。初めに、画像を 4×4 画素のブロックに分割する。ブロック内画素値を x_{ij} ($i, j = 1 \sim 4$)、ブロック内における平均値を L_A 、階調幅指標を L_D 、最大値を L_{\max} 、最小値を L_{\min} 、画素毎の量子化値を ϕ_{ij} ($i, j = 1 \sim 4$)、復号化値を y_{ij} ($i, j = 1 \sim 4$) とすると、符号化手順は図16に示すようになる。

【0004】図16の復号化手順に示すように、ブロック内における最大および最小画素値を求め、該最大値と最小値との間を4分割した一番上と下の部分に属する画素値の平均値を求め、これを Q_1, Q_2 とする。該 Q_1, Q_2 の平均値と差をそれぞれブロック内における平均値 L_A 、階調幅指標 L_D として定義する。

【0005】次に、ブロック内における平均値 L_A と階調幅指標 L_D によりブロック内における各画素値を4つに分類し、それぞれ2ビットの量子化値に量子化する。符号データは、6バイトで構成される。図17に示すよ

うに、第1バイトが平均値、第2バイトが階調幅指標、第3バイトから第6バイトが量子化値である。

【0006】また、復号化手順は、図18に示すように4種類の量子化値をブロック内における平均値 L_A 、階調幅指標 L_D により復号する。この固定長G B T C型符号をカラープリンタの画像メモリ回路上において適用した例を図19に示す。この符号化方式により、JPEG方式で劣化が目立つ文字やCG（コンピュータ・グラフィック）のような人工的な画像に対しても劣化の少ない圧縮が可能となる。

【0007】

【発明が解決しようとする課題】しかしながら、上記の従来における方法にあっては、ブロックの符号データを6バイトで構成するため、圧縮率は $3/8$ と効率的な圧縮は望めない。特に文字やCGのような人工的な画像にあっては、ブロック内が1つあるいは2つの画素値で構成されているブロックが多く、無駄な情報も符号データにしてしまうため、効率的な符号化ができないという問題点があった。

【0008】本発明は、上記に鑑みてなされたのものであって、従来方式の復号画像と同等の画質を保持し、従来のような無駄な符号データを少なくして符号化効率を向上させることを目的とする。

【0009】

【課題を解決するための手段】本発明は、上記の目的を達成するため、多値（カラー）画像情報をブロックに分割して、該ブロック毎に符号データを構成し、符号化する多値（カラー）画像の符号化装置において、前記ブロック内の画素値の構成情報を抽出し、前記ブロック内構成情報に応じて符号データの構成を変化させ、該符号データの一部であるブロック内の代表値を整数で割った際の余りの数により前記符号データの構成を切り換える制御手段を具備する符号化装置を提供するものである。

【0010】また、前記制御手段は、前記ブロック内の全画素値が同一の場合、先のブロック内情報も同一であるか探索し、該情報に応じて複数ブロックを連結して符号化するものである。

【0011】また、前記制御手段は、前記ブロック内が2種類の画素値で構成されている場合、画素値情報を符号化するための複数のブロック内量子化値構成パターンを示すコード番号と前記ブロック内の代表値により符号化するものである。

【0012】

【作用】本発明による符号化装置は、多値の画像情報をブロックに分割してブロック毎に符号化する際、ブロック内が1値あるいは2値で構成されているとき、従来のように決まった符号データを構成するのではなく、ブロック内における画素値の構成情報を応じて、符号データの構成をブロック内の代表値を整数で割った際の余りの数により切り換え、さらにブロック内量子化値構成パタ

ーンのコード化、複数ブロックの連結符号化を実行して、無駄な符号データを減少させることにより、符号長を増すことなく、効率的な圧縮を可能にする。

【0013】

【実施例】以下、本発明の一実施例を添付図面に基づいて具体的に説明する。図1は、本発明に係る多値（カラー）画像の符号化装置におけるシステム構成を示すブロック図である。図において、101は画像データを符号化装置102に入力させるホストコンピュータ、102は符号化装置であり、以下のように構成されている。すなわち、ホストコンピュータ101等から送信されるデータを受信する受信装置103、画像データを符号化するエンコーダ104、符号化された画像データを格納する画像メモリ105、符号化されたデータを復号化するデコーダ106、プリンタ108に画像データを送信する送信装置107とから構成されている。また、108は画像データを記録紙に印字出力するプリンタである。

【0014】次に、基本的な動作を説明する。本符号化装置は、スキャナ（画像読取装置）やファクシミリ受信装置、ホストコンピュータ101等から供給された多値（カラー）画像を受信装置103により受信して画像メモリ105に書き込む前にエンコーダ104により符号化し、また、画像メモリ105から読み出し、送信装置107を介してプリンタ108により印字出力する前にデコーダ106により復号化するシステムである。

【0015】また、図2は、図1に示した多値（カラー）の符号化装置の詳細構成を示すブロック図である。図において、この符号化装置102は、CPU201、ROM202、RAM203からなるマイクロコンピュータシステムと、画像メモリ105、キャッシュメモリ204、該キャッシュメモリ204を操作するキャッシュメモリ操作装置205、受信装置103、送信装置107とにより構成され、それらの機能は以下の通りである。

【0016】すなわち、CPU201は、ROM202内のプログラムおよびホストコンピュータ101等からのコマンドにより符号化装置102を制御する中央処理装置であり、ROM202は、CPU201が動作するための制御プログラムおよび画素値構成パターンコード（後述）等の固定データを格納している読み出し専用メモリであり、RAM203は、CPU201用のワークメモリ、入力データを格納するためのインプットバッファ等に使用するランダム・アクセス・メモリである。

【0017】また、画像メモリ105は、画像の符号データを書き込むためのランダム・アクセス・メモリであり、キャッシュメモリ204はブロック化された画像データを一時的に格納するランダム・アクセス・メモリである。受信装置103は、ホストコンピュータ101等から送信されるデータの受信を実行し、送信装置107は、実際に印字を行なうプリンタ108への画像イメー

ジデータの送信を実行する。

【0018】次に、動作について説明する。図3～図6は、図1に示したエンコーダ104による符号化処理の動作を示すフローチャートである。図3において、まず、ホストコンピュータ101等から送られてきた画像データの一部分（4096バイト）は一時的にRAM203に格納され、CPU201は画像データを4×4画素のブロックに分割し（S301）、キャッシュメモリ204にデータを格納する。さらに、CPU201は、ブロック内における最大画素値 L_{max} と最小画素値 L_{min} を求める（S302）。

【0019】その後、 $L_{max} = L_{min}$ であるか否かを判断し（S303）、この処理において、 $L_{max} = L_{min}$ であると判断した場合には、ブロック内における全画素値が同値であると判断し、その画素値を平均値LAとして記憶し（S304）、次のブロックのデータをキャッシュメモリ204に取り込み、同様に L_{max} と L_{min} を求める（S305）、 $L_{max} = L_{min}$ であるか否かを判断し（S306）。この処理において、 $L_{max} = L_{min}$ であると判断した場合には、次に、 $LA = L_{max}$ か否かを判断し（S307）、 $LA = L_{max}$ であると判断した場合には、カウント（ $i = i + 1$ ）して（S308）、上記ステップS304に戻る。

【0020】この処理において、ブロック内における画素値が前のブロックと同様の画素値を構成すると判断した場合には、ステップS304～ステップS308の処理ループを繰り返し、ブロック内の画素値がすべて同じであるブロックの数iを算出する。

【0021】この際、前記の符号データ構成であることを示すために、LAを整数3で割ったとき余りが0になるようにLAを変更する（S309）。図7に、この場合における符号データ構成を示す。その後、第1バイトをLA、第2バイトをiとして符号データを構成し、画像メモリ105に格納し（S310）、カウンタをリセット（ $i = 0$ ）する（S311）。

【0022】一方、上記ステップS303において、ブロック内がすべて同じ画素値ではないと判断した場合には、図4において、まず、ブロック内が画素値0と255の2値により構成されているか否かを判断し（S312、S313）、ブロック内が画素値0と255の2値で構成されていると判断した場合には、LAを3で割ったとき余りが1になる（LA=127）ようにする（S314）。次に、ブロック内の画素値を2値化する（S315）。そして、第1バイトをLA、第2、第3バイトを2値化したブロック内における全画素値として符号データを構成し、画像メモリ105に格納する（S316）。図8に、この場合における符号データ構成を示す。

【0023】次に、上記画素値構成以外の処理について説明する。上記ステップS312およびS313におい

て、キャッシュメモリ204にあるブロック内におけるデータがすべて同じ画素値あるいは0と255の2値により構成されていないと判断した場合は、CPU201は、最大値 L_{\max} と最小値 L_{\min} との間を4分割した一番上と下の部分に属する画素値の平均値 Q_1 、 Q_2 を求める(S319、S320)。その後、 Q_1 、 Q_2 の平均値と差をそれぞれブロック内における平均値 L_A 、階調幅指標 L_D として算出する(S321)。さらに、ブロック内におけるデータがすべて同じ画素値あるいは0と255の2値で構成されていないことを示すために、 L_A を整数3で割ったとき余りが2になるように L_A を変更する(S322)。

【0024】上記処理を実行した後、CPU201は、キャッシュメモリ204にあるブロック内における画素値データとRAM203にある L_{\max} 、 L_{\min} を比較して、ブロック内が2種類の画素値により構成されているか否かを判断する(S323)。このステップS323において、ブロック内が2種類の画素値で構成されないと判断した場合には、図5において、CPU201はブロック内における全画素を2値化($L_{\min} = 0$ 、 $L_{\max} = 1$)して(S324)、キャッシュメモリ204に格納する。

【0025】さらに、ブロック内における最初の量子化値が0であるか否かを判断し(S325)、このステップS325において、ブロック内における最初の値が0であると判断した場合には、ROM202に格納されている255通りの2値のブロック内画素値構成パターンコードを有するパターンコードテーブルAを選択し、パターンコードテーブルAの画素値構成とキャッシュメモリ204の2値化したデータを比較し、最も近いパターンを選択して、そのパターンのコード番号をRAM203に格納する(S326)。その後、ブロック内が2種類の画素値で構成され、パターンコードテーブルAにより符号化したことを示すために、 L_D を整数3で割ったとき余りが1になるように L_D を変更する(S327)。図9にパターンコードテーブルの一例を示す。

【0026】一方、ステップS325において、ブロック内における最初の画素値が0でないと判断した場合には、同様にパターンコードテーブルBによって符号化し(S329)、パターンコードテーブルBを選択したことを示すために、 L_D を整数3で割ったとき余りが2になるように L_D を変更する(S330)。

【0027】その後、ステップS327あるいはステップS330の処理を実行した後、第1バイトを L_A 、第2バイトを L_D 、第3バイトを適用パターンのコード番号として符号データを構成し、画像メモリ105に格納する(S328)。図10に、この場合における符号データ構成を示す。

【0028】また、上記ステップS323において、ブロック内における全画素が L_{\max} 、 L_{\min} により構成さ

れていないと判断した場合には、図6において、そのことを示すために L_D を整数3で割ったとき余りが0になるように L_D を変更する(S331)。次に、平均値 L_A と階調幅指標 L_D によりブロック内における各画素値を4つに分類し、それぞれ2ビットの量子化値に量子化する(S332、S333)。符号データは、6バイトで構成される。第1バイトを L_A 、第2バイトを L_D 、第3バイトから第6バイトを量子化値として符号データを構成し、画像メモリ105に格納する(S334)。図11に、この場合の符号データ構成を示す。

【0029】以上の処理を全ブロックが終了するまで繰り返す。すなわち、ステップS316、ステップS328、あるいはステップS334の処理が終了すると、次のブロックの処理へ移行し(S317)、全てのブロックが終了したか否かを判断して(S318)、全ブロックが終了していないと判断した場合には、上記ステップS302に戻り、一方、全ブロックが終了したと判断した場合には、本処理を終了する。

【0030】図12～図14は、図1に示したデコーダ106による復号化処理の動作を示すフローチャートである。図において、CPU201は、画像メモリ105に格納された符号データの第1バイト(L_A)を読み込み(S901)、さらに、その L_A を整数3で割り余りを求める(S902)。その後、余りが0であるか否かを判断し(S903)、この処理において、余りが0と判断した場合には、第2バイトのデータ(i)を読み込み(S904)、($i+1$)個のブロック内における全画素を $y_{i,j} = L_A$ で復号し(S905)、RAM203に該復号データを一時的に格納する。

【0031】一方、上記ステップS903において、余りが0ではないと判断した場合には、さらに、余りが1であるか否かを判断する(S906)。この処理において、 L_A を整数3で割った際の余りが1と判断した場合には、ブロック内における全画素を、

量子化値が0のとき $y_{i,j} = 0$ 、

量子化値が1のとき $y_{i,j} = 255$

により復号し(S907)、RAM203に該復号データを一時的に格納する。

【0032】また、上記ステップS906において、 L_A を整数3で割った際の余りが2の場合、余りが1でないと判断し、CPU201は、図13において、第2バイトのデータ(L_D)を読み込み(S908)、 L_D を整数3で割り余りを求める(S909)。次に、余りが0であるか否かを判断し(S910)、この処理において、余りが0であると判断した場合には、ブロック内における全画素を、

量子化値が01のとき $y_{i,j} = L_A - L_D / 2$ 、

量子化値が00のとき $y_{i,j} = L_A - L_D / 6$ 、

量子化値が10のとき $y_{i,j} = L_A + L_D / 6$ 、

量子化値が11のとき $y_{i,j} = L_A + L_D / 2$

により復号し(S911), RAM203に該復号データを一時的に格納する。

【0033】一方, 上記ステップS910において, 余りが0ではないと判断した場合には, 図14において, さらに余りが1であるか否かを判断する(S912)。この処理において, LDを整数3で割った際の余りが1と判断した場合には, CPU201は, ROM202にあるパターンコードテーブルAを選択し(S913), 第3バイトのデータ(コード番号)を読み込み(S914), そのコード番号に該当するブロック内における量子化パターンをパターンコードテーブルAから探索し, 画素毎の量子化値を決定する(S916)。

【0034】また, LDを整数3で割った際の余りが2と判断した場合には, ステップS912において, LAを整数3で割った際の余りが1でないと判断し, ROM202に格納されているパターンコードテーブルBを選択し(S915), 同様に画素毎の量子化値を決定する(S914, S916)。その後, ブロック内における全画素を,

量子化値が0のとき $y_{i,j} = LA - LD / 2$,

量子化値が1のとき $y_{i,j} = LA + LD / 2$

により復号し(S917), RAM203に該復号データを一時的に格納する。

【0035】次に, 上記ステップS905, ステップS907, ステップS911, あるいはステップS917を実行した後, 次のデータがあるか否かを判断し(S918), 次のデータが存在すると判断した場合には, 上記ステップS901に戻り, 次のデータがないと判断した場合には, 本処理を終了する。すなわち, CPU201は, RAM203に格納した復号データを4096バイト毎に送信装置107に送り, プリンタ108を介して出力する。CPU201は, 画像メモリ105に格納されている符号データがなくなるまで, 以上の処理を繰り返す。復号する符号データがなくなったとき, CPU201は, RAM203に格納した残りの復号データを送信装置107に送り, プリンタ108を介して出力する。

【0036】

【発明の効果】以上説明したように, 本発明による符号化装置によれば, 多値の画像情報をブロックに分割してブロック毎に符号化する際, ブロック内が1値あるいは2値で構成されているとき, ブロック内の画素値の構成情報を応じて, 符号データの構成をブロック内の代表値を整数で割った際の余りの数によって切り換える, さらにブロック内量子化値構成パターンのコード化, 複数ブロックの連結符号化を実行したため, 従来方式の復号画像と同等の画質を保ちながら, 無駄な符号データを少なくすることができ, 符号化効率を向上させることができる。

【図面の簡単な説明】

【図1】本発明に係る多値(カラー)画像の符号化装置のシステム構成を示すブロック図である。

【図2】本発明に係る符号化装置の概略構成を示すブロック図である。

【図3】本発明に係るエンコーダの符号化処理動作を示すフローチャートである。

【図4】本発明に係るエンコーダの符号化処理動作を示すフローチャートである。

【図5】本発明に係るエンコーダの符号化処理動作を示すフローチャートである。

【図6】本発明に係るエンコーダの符号化処理動作を示すフローチャートである。

【図7】本発明に係るブロック内が1種類の画素値により構成されている場合の符号データ構成を示す説明図である。

【図8】本発明に係るブロック内が0と255の画素値により構成されている場合の符号データ構成を示す説明図である。

【図9】本発明に係るパターンコードテーブルの構成を示す説明図である。

【図10】本発明に係るブロック内が2種類の画素値で構成されている場合の符号データ構成を示す説明図である。

【図11】ブロック内が3種類以上の画素値で構成されている場合の符号データ構成を示す説明図である。

【図12】本発明によるデコーダによる復号化処理動作を示すフローチャートである。

【図13】本発明によるデコーダによる復号化処理動作を示すフローチャートである。

【図14】本発明によるデコーダによる復号化処理動作を示すフローチャートである。

【図15】従来における符号化方法である固定長GBC型符号方式を示す説明図である。

【図16】従来における固定長GBC型符号方式の符号化手順を示す説明図である。

【図17】従来における固定長GBC型符号方式の符号データを示す説明図である。

【図18】従来における固定長GBC型符号方式の復号化手順を示す説明図である。

【図19】従来における固定長GBC型符号方式をカラープリンタの画像メモリ回路上に適用した例を示す説明図である。

【符号の説明】

102 符号化装置
103 受信装置

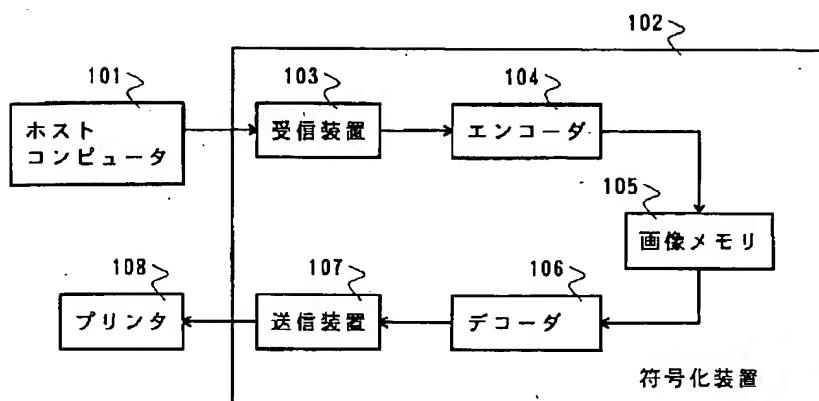
104 エンコーダ
105 画像メモリ

106 デコーダ
107 送信装置

50 201 CPU
202 R

OM
203 RAM* ヤッッシュメモリ
204 キ* 205 キャッシュメモリ操作装置

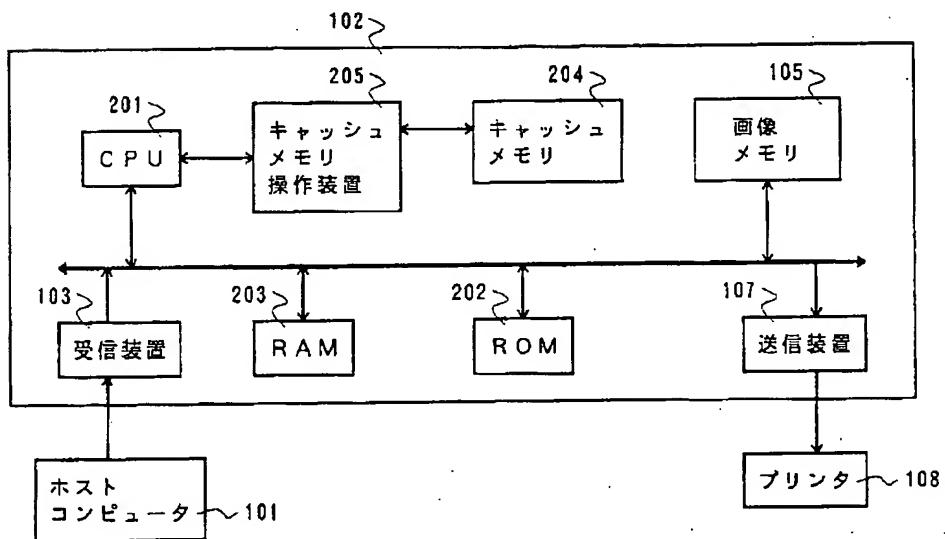
【図1】



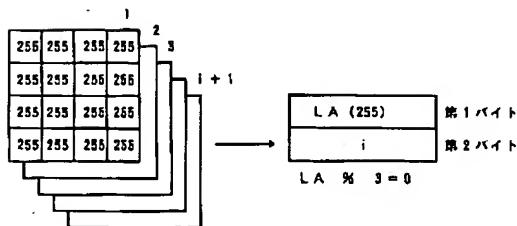
【図18】

第1バイト	LA
第2バイト	LD
第3バイト	$\phi_{ij} = (i, j = 1 \sim 4)$
第4バイト	
第5バイト	1画素毎の量子化係数は2ビット
第6バイト	

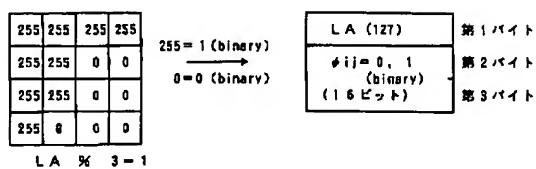
【図2】



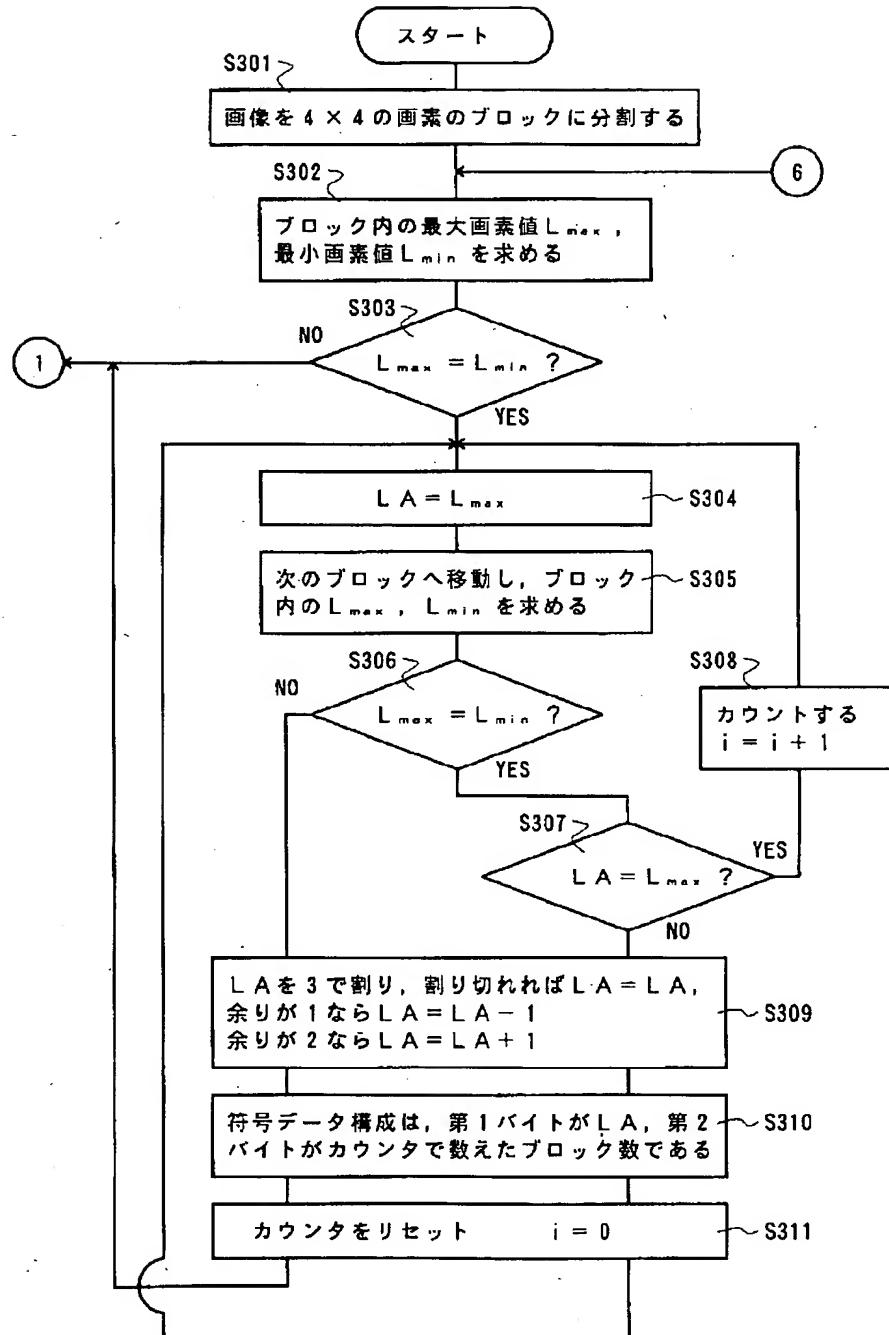
【図7】



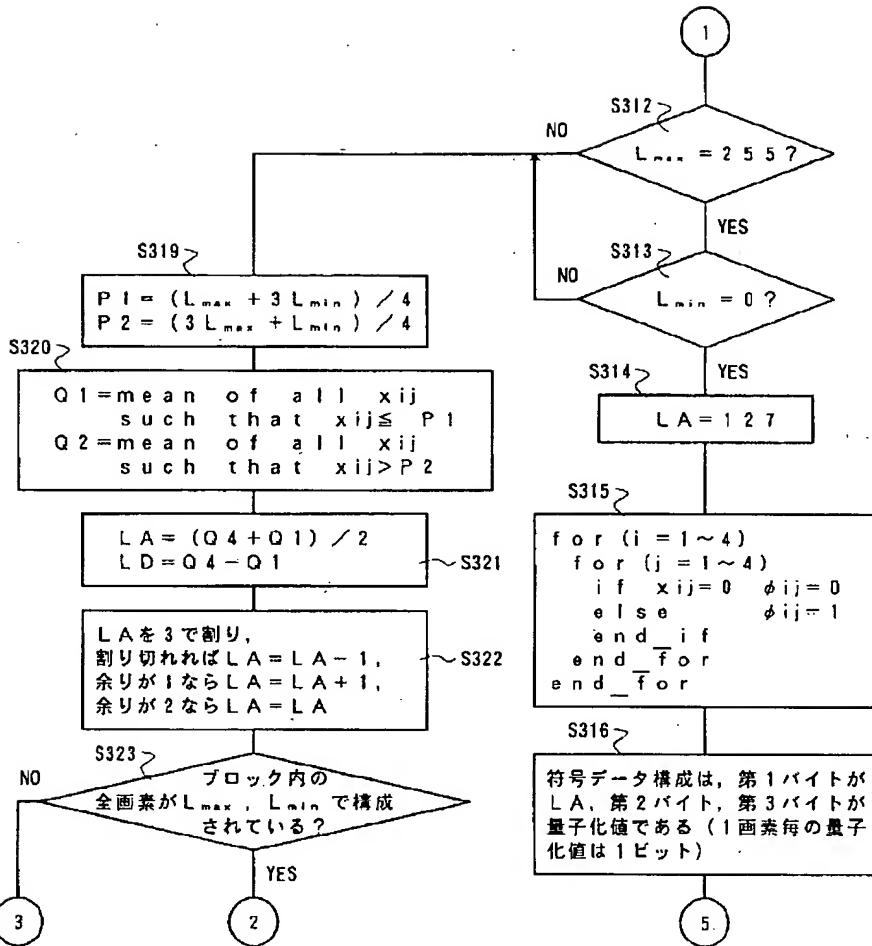
【図8】



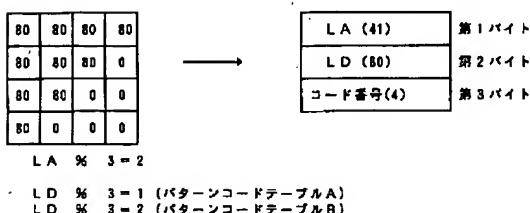
【図3】



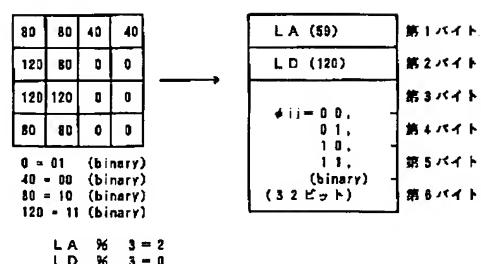
【図4】



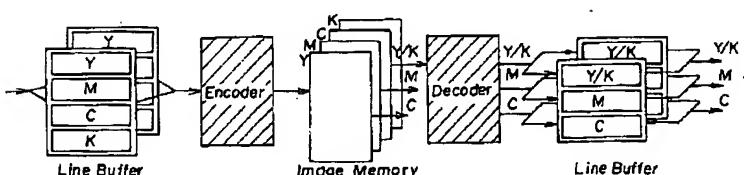
【図10】



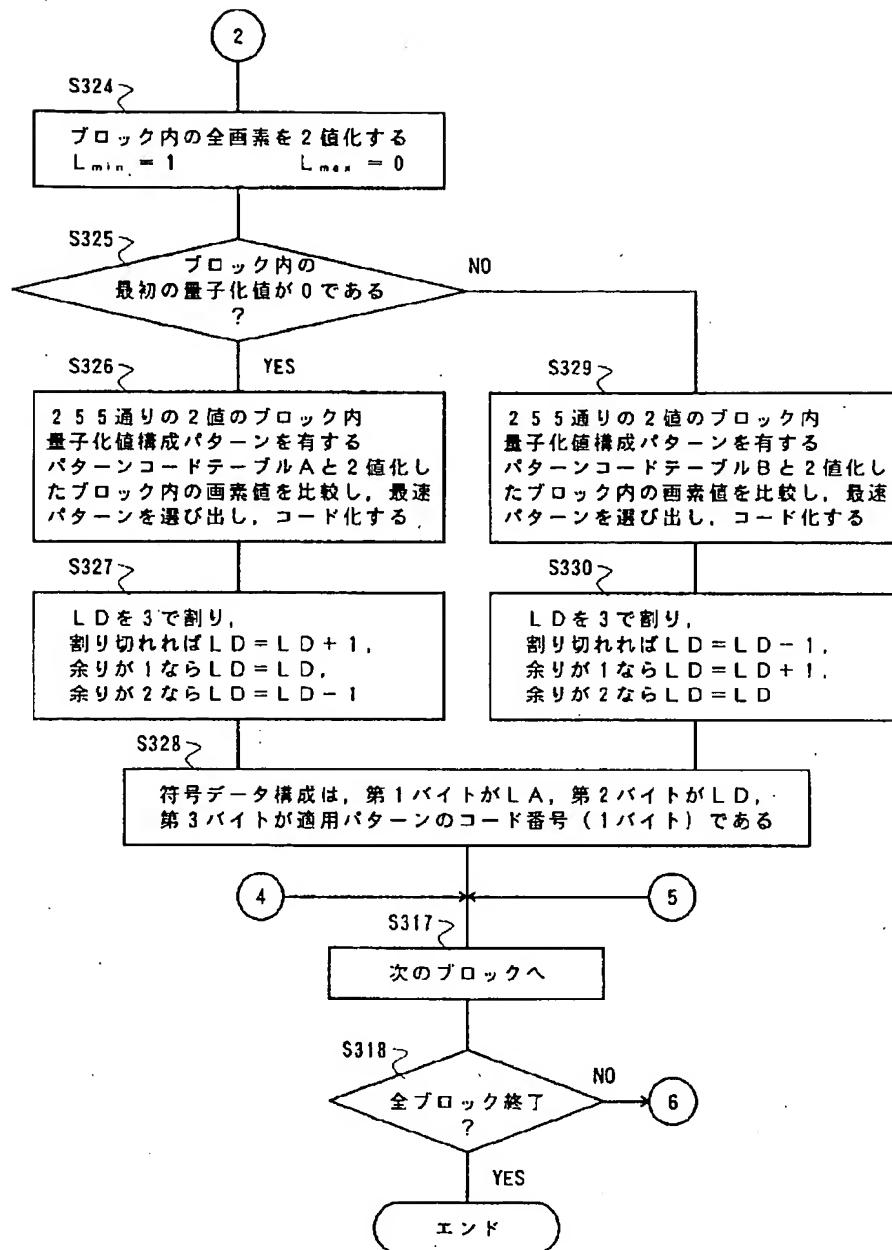
【図11】



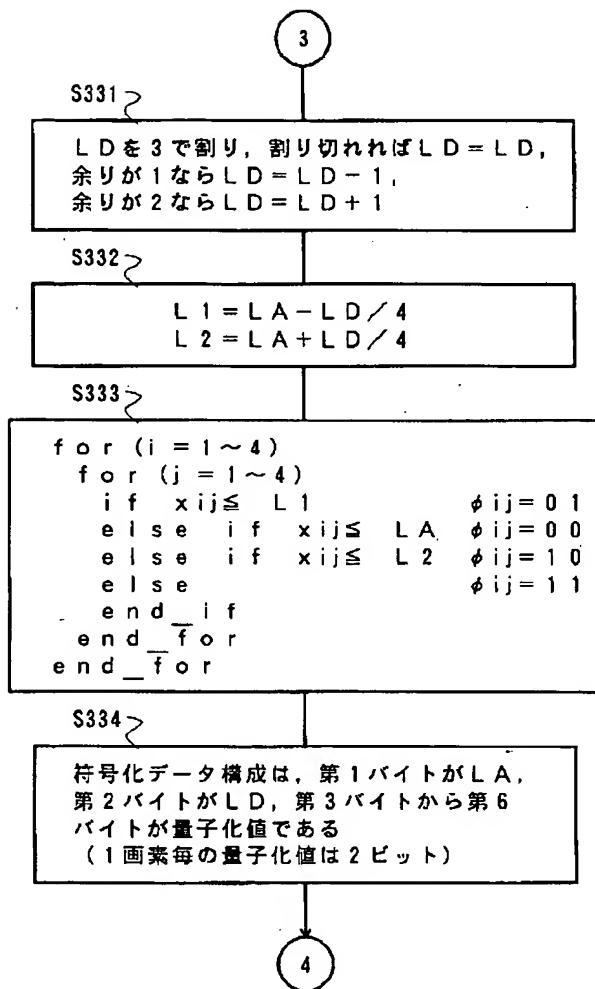
【図19】



【図5】



【図6】

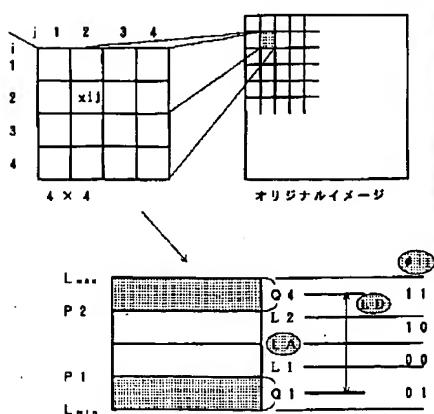


【図9】

0	1	2	3
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0	1 1 1 1 1 1 1 0 1 1 0 0 1 0 0 0
4	...	254	255
パターンコードテーブルA			

0	1	...	255
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1	...	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
0	1	...	255
パターンコードテーブルB			

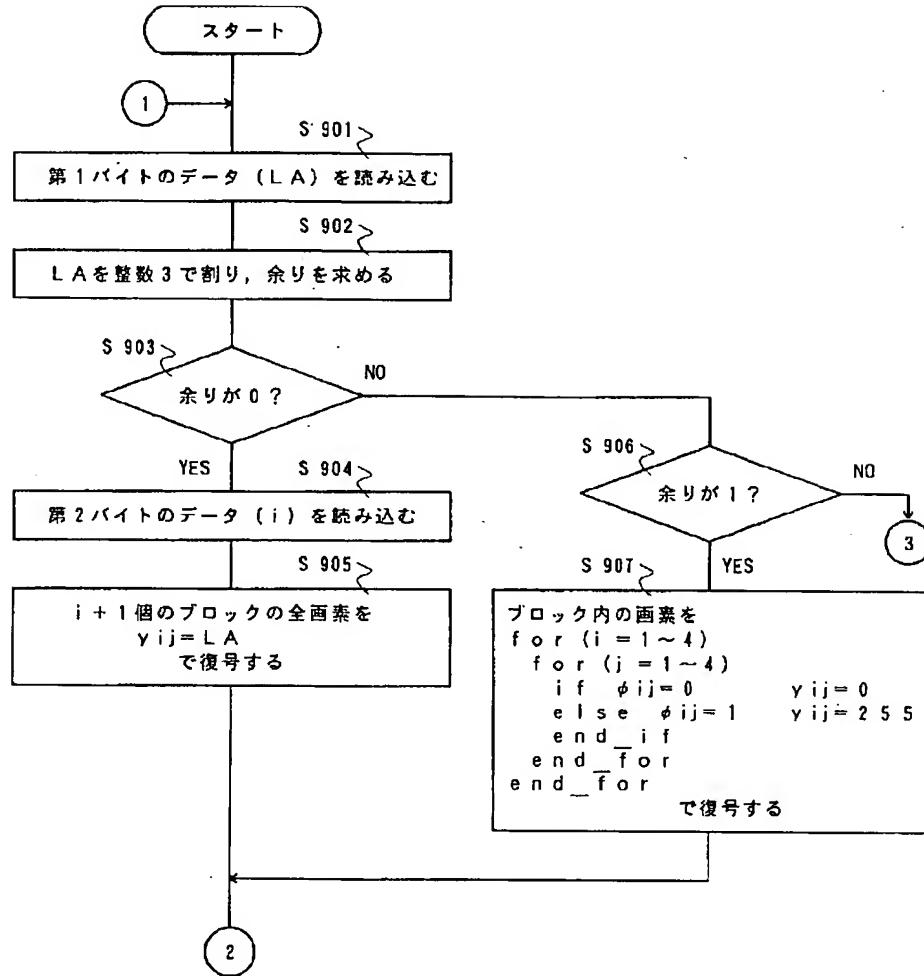
【図15】



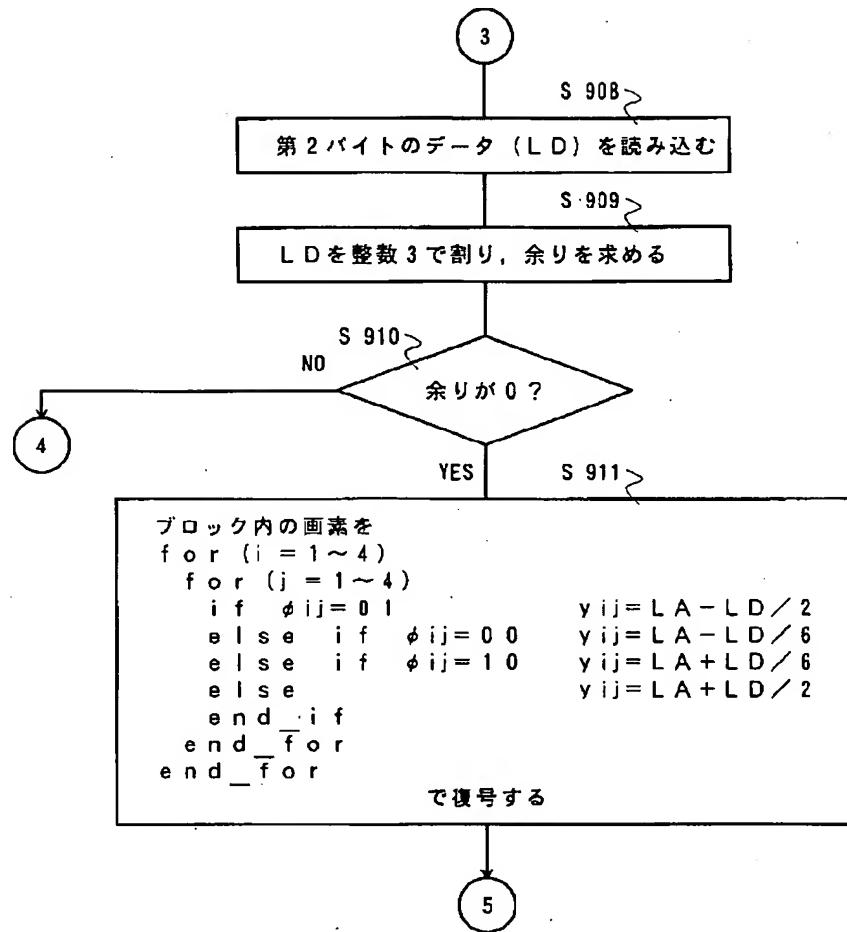
【図17】

for (i = 1, . . . , 4)	
for (j = 1, . . . , 4)	
if φij = 01	yij = LA - LD/2
else if φij = 00	yij = LA - LD/6
else if φij = 10	yij = LA + LD/6
else	yij = LA + LD/2
end_if	
end_for	
end_for	

【图12】



【図13】

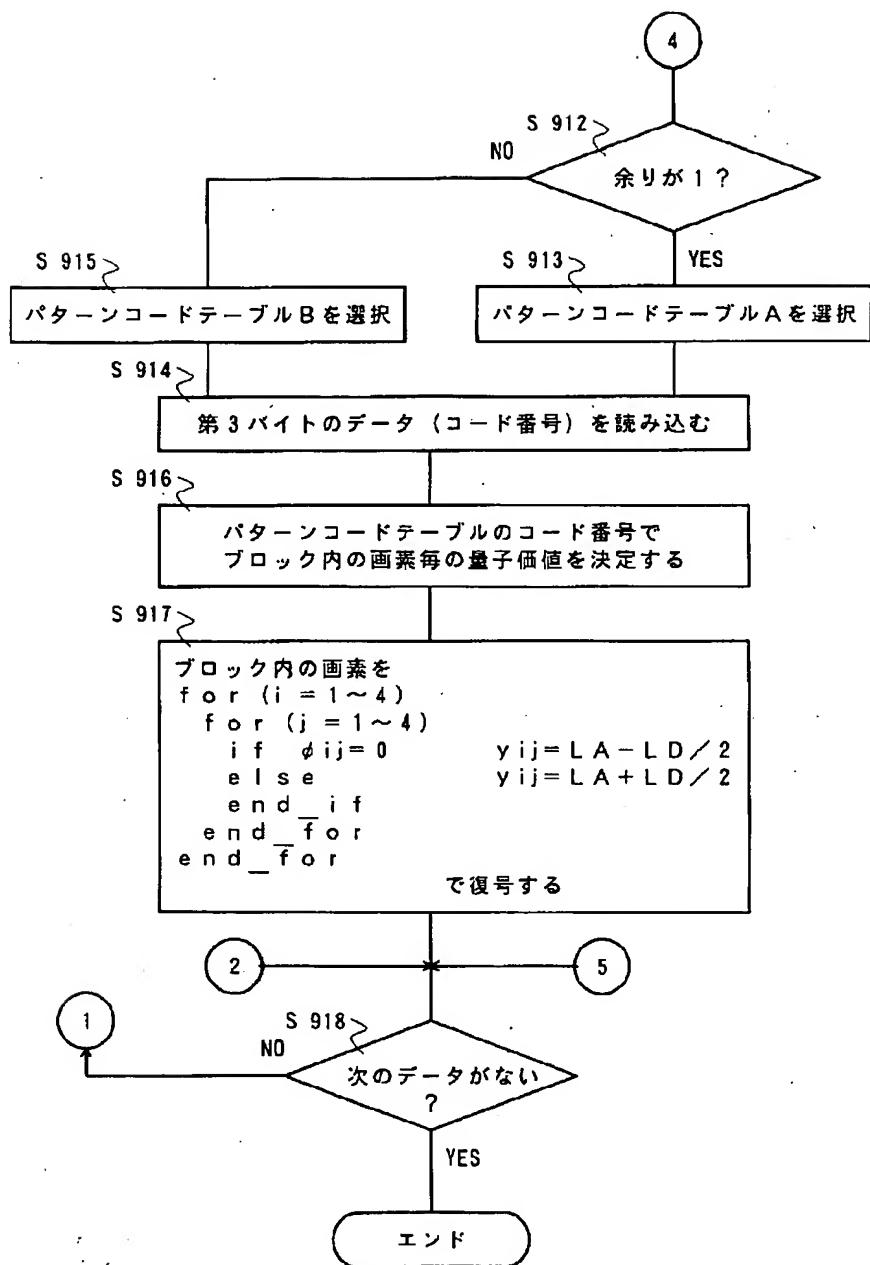


【図16】

```

P1 = (L_max + 3 L_min) / 4
P2 = (3 L_max + L_min) / 4
Q1 = mean of all xij such that xij ≤ P1
Q4 = mean of all xij such that xij > P2
LA = (Q1 + Q4) / 2
LD = Q4 - Q1
L1 = LA - LD / 4
L2 = LA + LD / 4
for (i = 1, . . . , 4)
    for (j = 1, . . . , 4)
        if xij ≤ L1 phi_ij = 01 (binary)
        else if xij ≤ LA phi_ij = 00 (binary)
        else if xij ≤ L2 phi_ij = 10 (binary)
        else phi_ij = 11 (binary)
    end_for
end_for
    
```

【図14】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-334870

(43)Date of publication of application : 02.12.1994

(51)Int.Cl. H04N 1/41

G06F 15/66

(21)Application number : 05-141524 (71)Applicant : RICOH CO LTD

(22)Date of filing : 19.05.1993 (72)Inventor : KOMATSU MANABU

(54) CODER

(57)Abstract:

PURPOSE: To improve the coding efficiency by changing structure of coding data in response to extraction structure information of a picture element value in a block and selecting a structure of the code data based on a remainder dividing a representative value in the block by an integer.

CONSTITUTION: Part of set picture data is tentatively stored in a RAM 203 and a CPU 201 divides picture data into 4×4 picture elements and stores it into a memory 204 and to obtain a maximum picture element Lmax and a minimum picture element Lmin in the block. When the equality is in existence in the processing of $L_{max}=L_{min}$, all picture elements are used for the same value and the picture element is stored as a mean value LA and number (i) of blocks in which the picture elements in the block are the same is calculated. In this case, the LA is revised by setting a remainder when the mean value LA is divided by an integer 3 is zero. Then code data are formed in which a

1st byte is set to LA and a 2nd byte is set to (i) and stored in a picture memory 105. The similar processing is executed in the other case.

LEGAL STATUS [Date of request for examination] 19.01.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3260910

[Date of registration] 14.12.2001

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

JPO and INPIT are not responsible for any

damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the coding equipment of the multiple-value (color) image which divides multiple-value (color) image information into a block, constitutes code data for this every block, and is encoded With too much number at the time of extracting the configuration information of the pixel value within said block, changing the configuration of code data according to the configuration information within said block, and breaking integrally the central value within the block which is this a part of code data Coding equipment characterized by providing the control means which switches the configuration of said code data.

[Claim 2] Said control means is coding equipment according to claim 1 characterized by whether the information within a block on previous is also the same, and searching, connecting two or more blocks and encoding according to this information when the total pixel value within said block is the same.

[Claim 3] Said control means is coding equipment according to claim 1

characterized by encoding with the code number which shows two or more quantization value configuration patterns within a block for encoding pixel value information, and the central value within said block when the inside of said block consists of two kinds of pixel values.



DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention is used for digital color picture processing, divides multiple-value (color) image information into a block, constitutes code data for this every block, and relates to the coding equipment of the multiple-value (color) image to encode.

[0002]

[Description of the Prior Art] Conventionally, the color picture information on a multiple value is divided into a block, and code data is constituted for this every block, it is the coding approach of the color picture to encode, and even if it is a color picture with high spatial frequency other than a natural image, the fixed-length GBTC (Generalized Block TruncationCoding) mold coding method is learned as the coding approach with the sufficient quality of a decode image.

[0003] Drawing 15 is the explanatory view showing a fixed-length GBTC mold sign method. Introduction and an image are divided into a 4x4-pixel block. the average [value / within a block / pixel] within x_{ij} ($i, j=1-4$) and a block -- when LA

and a gradation width-of-face index are set to LD and ϕ_{ijj} ($i, j=1-4$) and a decryption value are set [maximum / Lmax and the minimum value] to y_{ij} ($i, j=1-4$) for Lmin and the quantization value for every pixel, a coding procedure comes to be shown in drawing 16 .

[0004] As shown in the decryption procedure of drawing 16 , the max and the minimum pixel value within a block are calculated, the average of the pixel value belonging to the top which quadrisected between this maximum and the minimum values, and a lower part is calculated, and this is set to Q1 and Q2. this -- the average value and difference of Q1 and Q2 are defined as the average value LA within a block, and a gradation width-of-face index LD, respectively.

[0005] Next, each pixel value within a block is classified into four according to the average value LA and the gradation width-of-face index LD within a block, and it quantizes in quantization value of 2 bits, respectively. Code data consists of 6 bytes. As shown in drawing 17 , for the 1st byte, an average value and the 2nd byte is [the 6th byte] quantization values from a gradation width-of-face index and the 3rd byte.

[0006] Moreover, a decryption procedure is decoded with the average LA within a block of four kinds of quantization values, and the gradation width-of-face

index LD, as shown in drawing 18 . The example which applied this fixed-length GBTC mold sign on the image memory circuit of a color printer is shown in drawing 19 . With this coding method, little compression of degradation is attained also to an artificial image like the alphabetic character in which degradation is conspicuous, or CG (computer graphic) by the JPEG method.

[0007]

[Problem(s) to be Solved by the Invention] However, if it is in the approach in the above-mentioned former, since the code data of a block is constituted from 6 bytes, compression with as effective compressibility as three eighths cannot be desired. If it was in an artificial image like especially an alphabetic character and CG, there were many blocks with which the inside of a block consists of one or two pixel values, and in order to also make useless information into code data, there was a trouble that efficient coding could not be performed.

[0008] This invention is a making-in view of the above thing, holds image quality equivalent to the decode image of the conventional method, and aims at losing useless code data like before few, and raising coding effectiveness.

[0009]

[Means for Solving the Problem] In the coding equipment of the multiple-value

(color) image which it divides multiple-value (color) image information into a block, and constitutes code data for this every block in order that this invention may attain the above-mentioned purpose, and is encoded With too much number at the time of extracting the configuration information of the pixel value within said block, changing the configuration of code data according to the configuration information within said block, and breaking integrally the central value within the block which is this a part of code data The coding equipment possessing the control means which switches the configuration of said code data is offered.

[0010] Moreover, when the total pixel value within said block is the same, said control means searches for whether it is the same, and according to this information, the information within a block on previous also connects two or more blocks, and it encodes it.

[0011] Moreover, said control means is encoded with the code number which shows two or more quantization value configuration patterns within a block for encoding pixel value information, and the central value within said block, when the inside of said block consists of two kinds of pixel values.

[0012]

[Function] The coding equipment by this invention does not constitute code data regular like before, when the inside of a block consists of one value or binary, in case the image information of a multiple value is divided into a block and it encodes for every block. According to the configuration information of the pixel value within a block, with too much number at the time of breaking the central value within a block of the configuration of code data integrally Efficient compression is enabled without increasing code length a switch and by performing coding of the quantization value configuration pattern within a block, and two or more blocks connection coding further, and decreasing useless code data.

[0013]

[Example] Hereafter, one example of this invention is concretely explained based on an accompanying drawing. Drawing 1 is the block diagram showing the system configuration in the coding equipment of the multiple-value (color) image concerning this invention. In drawing, the host computer which 101 makes input image data into coding equipment 102, and 102 are coding equipment, and are constituted as follows. That is, it consists of the receiving set 103 which receives the data transmitted from host computer 101 grade, the

encoder 104 which encodes image data, an image memory 105 which stores the encoded image data, a decoder 106 which decrypts the encoded data, and a sending set 107 which transmits image data to a printer 108. Moreover, 108 is a printer which carries out the printout of the image data to the recording paper.

[0014] Next, fundamental actuation is explained. before it encodes with an encoder 104 before a receiving set's 103 receiving a scanner (image reader), a facsimile receiving set, and the multiple-value (color) image supplied from the host computer 101 grade and writing in an image memory 105, and it reads this coding equipment from an image memory 105 and it carries out a printout by the printer 108 through a sending set 107, it is a decrypted system which comes out of by the decoder 106.

[0015] Moreover, drawing 2 is the block diagram showing the detail configuration of the coding equipment of a multiple value (color) shown in drawing 1 . In drawing, this coding equipment 102 is constituted by the microcomputer system which consists of CPU201, ROM202, and RAM203, and the cache memory operating set 205 which operates an image memory 105, cache memory 204, and this cache memory 204, a receiving set 103 and a sending set 107, and those functions are as follows.

[0016] That is, CPU201 is a central processing unit which controls coding equipment 102 by the command from the program and host computer 101 grade in ROM202, ROM202 is a read-only memory which stores fixed data, such as a control program for CPU201 to operate and a pixel value configuration pattern code (after-mentioned), and RAM203 is random access memory used for the input buffer for storing the work-piece memory for CPU201, and input data etc.

[0017] Moreover, an image memory 105 is the random access memory for writing in the code data of an image, and cache memory 204 is random access memory which stores the blocked image data temporarily. A receiving set 103 performs reception of the data transmitted from host computer 101 grade, and a sending set 107 performs transmission of the image image data to the printer 108 which actually prints.

[0018] Next, actuation is explained. Drawing 3 - drawing 6 are flow charts which show actuation of the coding processing by the encoder 104 shown in drawing 1. In drawing 3 , first, a part of image data (4096 bytes) sent from the host computer 101 grade is temporarily stored in RAM203, CPU201 divides image data into a 4x4-pixel block (S301), and it stores data in cache memory 204. Furthermore, CPU201 is the maximum pixel value Lmax within a block. The

minimum pixel value Lmin It asks (S302).

[0019] then, Lmax =Lmin it is -- a ***** -- judging (S303) -- this processing -- setting -- Lmax =Lmin it is -- **, when it judges Judge that the total pixel value within a block is equivalent, and the pixel value is memorized as the average LA (S304). the data of the following block -- cache memory 204 -- incorporating -- the same -- Lmax Lmin -- asking (S305) -- Lmax =Lmin it is -- a ***** -- judging (S306) . this processing -- setting -- Lmax =Lmin it is -- ** -- the case where it judges next -- LA=Lmax ***** -- judging (S307) -- LA=Lmax it is -- ** -- when it judges, it counts (i=i +1) (S308), and returns to the above-mentioned step S304.

[0020] It sets to this processing, and when it is judged that the pixel value within a block constitutes the same pixel value as a pre- block, the processing loop formation of step S304 - step S308 is repeated, and all the pixel values within a block compute several i of the same block.

[0021] Under the present circumstances, in order to show that it is the aforementioned code data configuration, when LA is divided by the integer 3, LA is changed so that remainder may be set to 0 (S309). The code data configuration which can be set in this case to drawing 7 is shown. Then, code data is constituted using the 2nd byte as i by using the 1st byte to LA, it stores in

an image memory 105 (S310), and a counter is reset (S311). (i= 0)

[0022] On the other hand, when it is judged in the above-mentioned step S303 that all the inside of a block is not the same pixel values In drawing 4 , it judges first whether the inside of a block is constituted by binary [of the pixel values 0 and 255] (S312, S313). remainder is set to 1 when it is judged that the inside of a block consists of binary [of the pixel values 0 and 255], and LA is divided by 3 (LA=127) -- it is made like (S314). Next, the pixel value within a block is made binary (S315). And code data is constituted as a total pixel value within the block which made LA, the 2nd, and the 3rd byte binary for the 1st byte, and it stores in an image memory 105 (S316). The code data configuration which can be set in this case to drawing 8 is shown.

[0023] Next, processings other than the above-mentioned pixel value configuration are explained. the above-mentioned step S -- the case where it is judged that all the data within the block in cache memory 204 are not constituted by the same pixel value or binary [of 0 and 255] in 312 and 313 -- CPU201 -- maximum Lmax The minimum value Lmin The averages Q1 and Q2 of the pixel value belonging to the top which quadrisected between, and a lower part are calculated (S319, S320). Then, the average value and difference of Q1 and Q2

are computed as the average value LA within a block, and a gradation width-of-face index LD, respectively (S321). Furthermore, in order to show that all the data within a block do not consist of same pixel value or binary [of 0 and 255], when LA is divided by the integer 3, LA is changed so that remainder may be set to 2 (S322).

[0024] CPU201 after performing the above-mentioned processing is Lmax in the pixel value data and RAM203 within the block in cache memory 204, and Lmin. It judges whether it compares and the inside of a block is constituted by two kinds of pixel values (S323). In this step S323, when it is judged that the inside of a block consists of two kinds of pixel values, it sets to drawing 5 , and CPU201 makes all the pixels within a block binary (Lmin =0, Lmax =1) (S324), and stores them in cache memory 204.

[0025] Furthermore, judge whether the first quantization value within a block is 0 (S325), and it sets to this step S325. When it is judged that the first value within a block is 0 The data which chose the pattern code table A which has the pixel value configuration pattern code within a block binary [255 kinds of] stored in ROM202, and the pixel value configuration of the pattern code table A and cache memory 204 made binary It compares, the nearest pattern is chosen and

the code number of the pattern is stored in RAM203 (S326). Then, the inside of a block consists of two kinds of pixel values, and in order to show having encoded with the pattern code table A, when LD is divided by the integer 3, LD is changed so that remainder may be set to 1 (S327). An example of a pattern code table is shown in drawing 9.

[0026] On the other hand, it sets to step S325, when it is judged that the first pixel value within a block is not 0, it encodes with the pattern code table B similarly (S329), and in order to show having chosen the pattern code table B, when LD is divided by the integer 3, LD is changed so that remainder may be set to 2 (S330).

[0027] Then, after performing processing of step S327 or step S330, LA and the 2nd byte are constituted for the 1st byte, code data is constituted for LD and the 3rd byte as a code number of an application pattern, and it stores in an image memory 105 (S328). The code data configuration which can be set in this case to drawing 10 is shown.

[0028] Moreover, the all [set to the above-mentioned step S323, and] pixel within a block is Lmax and Lmin. When it is judged that it is not constituted, in order to show that, when LD is divided by the integer 3, in drawing 6, LD is

changed so that remainder may be set to 0 (S331). Next, each pixel value within a block is classified into four according to an average value LA and the gradation width-of-face index LD, and it quantizes in quantization value of 2 bits, respectively (S332, S333). Code data consists of 6 bytes. Code data is constituted using the 6th byte as a quantization value for the 2nd byte from LD and the 3rd byte by using the 1st byte to LA, and it stores in an image memory 105 (S334). The code data configuration in this case is shown in drawing 11.

[0029] It repeats until a whole block ends the above processing. That is, after processing of step S316, step S328, or step S334 is completed, when it shifts to processing of the following block (S317), it judges whether all blocks were completed (S318) and it is judged that the whole block is not completed, return is ended to the above-mentioned step S302, and this processing is ended when it is judged on the other hand that the whole block was completed.

[0030] Drawing 12 - drawing 14 are flow charts which show actuation of the decryption processing by the decoder 106 shown in drawing 1. In drawing, CPU201 reads the 1st byte (LA) of the code data stored in the image memory 105 (S901), further, divides the LA by the integer 3, and asks for remainder (S902). Then, it judges whether remainder is 0 (S903), and when remainder

judges it as 0 in this processing, the 2nd byte of data (i) are read (S904), all the pixels within the block of an individual (i+1) are decoded by $y_{ij}=LA$ (S905), and these decode data are temporarily stored in RAM203.

[0031] On the other hand, when it is judged in the above-mentioned step S903 that remainder is not 0, it judges further whether remainder is 1 (S906). In this processing, when the remainder at the time of dividing LA by the integer 3 judges it as 1, all the pixels within a block are decoded by $y_{ij}=255$, when a quantization value is 0 and $y_{ij}=0$ and a quantization value are 1 (S907), and these decode data are temporarily stored in RAM203.

[0032] Moreover, in the above-mentioned step S906, when the remainder at the time of dividing LA by the integer 3 is 2, it judges that remainder is not 1, and in drawing 13, CPU201 reads the 2nd byte of data (LD) (S908), divides LD by the integer 3, and asks for remainder (S909). next, when it judges whether remainder is 0 (S910) and it is judged in this processing that remainder is 0 All the pixels within a block are decoded by $y_{ij}=LA+LD/2$, when a quantization value is 01, $y_{ij}=LA-LD/2$ and a quantization value are 00, $y_{ij}=LA-LD/6$ and a quantization value are 10 and $y_{ij}=LA+LD/6$ and a quantization value are 11 (S911). These decode data are temporarily stored in RAM203.

[0033] On the other hand, when it is judged in the above-mentioned step S910 that remainder is not 0, in drawing 14, it judges whether remainder is 1 further (S912). In this processing, when the remainder at the time of dividing LD by the integer 3 judges it as 1, CPU201 chooses the pattern code table A in ROM202 (S913), reads the 3rd byte of data (code number) (S914), looks for the quantization value pattern within the block applicable to that code number from the pattern code table A, and determines the quantization value for every pixel (S916).

[0034] Moreover, when the remainder at the time of dividing LD by the integer 3 judges it as 2, in step S912, it judges that the remainder at the time of dividing LA by the integer 3 is not 1, the pattern code table B stored in ROM202 is chosen (S915), and the quantization value for every pixel is determined similarly (S914, S916). Then, all the pixels within a block are decoded by $y_{ij}=LA+LD/2$, when a quantization value is 0 and $y_{ij}=LA-LD/2$ and a quantization value are 1 (S917), and these decode data are temporarily stored in RAM203.

[0035] Next, this processing is ended when it is judged that there are not return and the following data in the above-mentioned step S901 when it judges whether there are any following data (S918) and it is judged that the following data exist

after performing the above-mentioned step S905, step S907, step S911, or step S917. That is, CPU201 minds the decode data stored in RAM203 every 4096 bytes, minds [107] delivery and a printer 108, and outputs. CPU201 repeats the above processing until the code data stored in the image memory 105 is lost. When the code data to decode is lost, CPU201 outputs the remaining decode data stored in RAM203 to a sending set 107 through delivery and a printer 108.

[0036]

[Effect of the Invention] As explained above, in case according to the coding equipment by this invention the image information of a multiple value is divided into a block and it encodes for every block, when the inside of a block consists of one value or binary, According to the configuration information of the pixel value within a block, with too much number at the time of breaking the central value within a block of the configuration of code data integrally Maintaining image quality equivalent to the decode image of the conventional method, since coding of the quantization value configuration pattern within a block and two or more blocks connection coding were performed further, a switch and, useless code data can be lessened and coding effectiveness can be raised.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the system configuration of the coding equipment of the multiple-value (color) image concerning this invention.

[Drawing 2] It is the block diagram showing the outline configuration of the

coding equipment concerning this invention.

[Drawing 3] It is the flow chart which shows coding processing actuation of the encoder concerning this invention.

[Drawing 4] It is the flow chart which shows coding processing actuation of the encoder concerning this invention.

[Drawing 5] It is the flow chart which shows coding processing actuation of the encoder concerning this invention.

[Drawing 6] It is the flow chart which shows coding processing actuation of the encoder concerning this invention.

[Drawing 7] It is the explanatory view showing a code data configuration when the inside of the block concerning this invention is constituted by one kind of pixel value.

[Drawing 8] It is the explanatory view showing a code data configuration when the inside of the block concerning this invention is constituted by the pixel value of 0 and 255.

[Drawing 9] It is the explanatory view showing the configuration of the pattern code table concerning this invention.

[Drawing 10] It is the explanatory view showing a code data configuration when

the inside of the block concerning this invention consists of two kinds of pixel values.

[Drawing 11] It is the explanatory view showing a code data configuration when the inside of a block consists of three or more kinds of pixel values.

[Drawing 12] It is the flow chart which shows the decryption processing actuation by the decoder by this invention.

[Drawing 13] It is the flow chart which shows the decryption processing actuation by the decoder by this invention.

[Drawing 14] It is the flow chart which shows the decryption processing actuation by the decoder by this invention.

[Drawing 15] It is the explanatory view showing the fixed-length GBTC mold sign method which is the coding approach in the former.

[Drawing 16] It is the explanatory view showing the coding procedure of the fixed-length GBTC mold sign method in the former.

[Drawing 17] It is the explanatory view showing the code data of the fixed-length GBTC mold sign method in the former.

[Drawing 18] It is the explanatory view showing the decryption procedure of the fixed-length GBTC mold sign method in the former.

[Drawing 19] It is the explanatory view showing the example which applied the fixed-length GBTC mold sign method in the former on the image memory circuit of a color printer.

[Description of Notations]

102 Coding Equipment 103 Receiving Set

104 Encoder 105 Image Memory

106 Decoder 107 Sending Set

201 CPU 202 ROM

203 RAM 204 Cache Memory

205 Cache Memory Operating Set